

# M-7065

---

## *User Manual*

---

### **Warranty**

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

### **Warning!**

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, not for any infringements of patents or other rights of third parties resulting from its use.

### **Copyright**

Copyright @ 2007 by ICP DAS Co., Ltd. All rights are reserved.

### **Trademark**

The names used for identification only may be registered trademarks of their respective companies.

## Table of Contents

<b>1. Introduction .....</b>	<b>5</b>
1.1 <i>More Information</i> .....	6
1.2 <i>Pin Assignment</i> .....	7
1.3 <i>Specifications</i> .....	8
1.4 <i>Block Diagram</i> .....	9
1.5 <i>Wire Connection</i> .....	10
1.6 <i>Quick Start</i> .....	11
1.7 <i>Default Settings</i> .....	16
1.8 <i>Configuration Tables</i> .....	16
1.9 <i>DI/O Data Format Table</i> .....	17
<b>2. DCON Protocol .....</b>	<b>18</b>
2.1 %AANNTTCCFF.....	22
2.2 #**.....	24
2.3 #AA00( <i>Data</i> ).....	26
2.4 #AA0A( <i>Data</i> ) .....	29
2.5 #AA1cDD.....	32
2.6 #AAAcDD .....	34
2.7 #AAN.....	36
2.8 \$AA2.....	38
2.9 \$AA4.....	40
2.10 \$AA5.....	43
2.11 \$AA6.....	45
2.12 \$AAC .....	47
2.13 \$AACN.....	49
2.14 \$AAF .....	51
2.15 \$AALS.....	53
2.16 \$AAM.....	56

2.17 \$AAP .....	58
2.18 \$AAPN.....	60
2.19 @AA.....	62
2.20 @AA( <i>Data</i> ).....	64
2.21 ~AAO( <i>Name</i> ).....	66
2.22 ~AAD.....	68
2.23 ~AADVV.....	70
2.24 ~AAI.....	72
2.25 ~AATnn.....	75
2.26 ~**.....	76
2.27 ~AA0.....	77
2.28 ~AA1.....	80
2.29 ~AA2.....	82
2.30 ~AA3EVV .....	84
2.31 ~AA4V .....	86
2.32 ~AA5V .....	88
2.33 ~AARDvv.....	90
2.34 ~AARD.....	92
3. Modbus RTU Protocol .....	94
3.1 01(0x01) Read coils.....	97
3.2 02(0x02) Read discrete inputs.....	101
3.3 03(0x03) Read multiple registers .....	103
3.4 04(0x04) Read multiple registers .....	107
3.5 05(0x05) Write single coils .....	110
3.6 06(0x06) Write multiple registers.....	114
3.7 15(0x0F) Write multiple coils .....	116
3.8 46(0x46) Read/Write module settings .....	119
3.8.1 00(0x00) Read Module Name .....	120

<i>3.8.2 04(0x04) Set Module Address .....</i>	121
<i>3.8.3 05(0x05) Read Communication.....</i>	122
<i>3.8.4 06(0x06) Set Communication .....</i>	124
<i>3.8.5 32(0x20) Read Firmware Version .....</i>	126
<i>3.8.6 33(0x21) Set Digital Input Count Edge.....</i>	127
<i>3.8.7 34(0x22) Read Digital Input Count Edge .....</i>	129
<i>3.8.8 39(0x27) Set Power-on Value .....</i>	131
<i>3.8.9 40(0x28) Read Power-on Value .....</i>	133
<i>3.8.10 40(0x29) Set DI/O Active Status.....</i>	134
<i>3.8.11 42(0x2A) Read DI/O Active Status .....</i>	136
<i>3.8.12 53(0x35) Read the Response Delay Time .....</i>	138
<i>3.8.13 54(0x36) Set the Response Delay Time .....</i>	140
<i>4. Modbus ASCII Protocol.....</i>	141
<i>A. Appendix.....</i>	143
<i>A.1 INIT Mode.....</i>	143
<i>A.2 Dual Watchdog operation.....</i>	144
<i>A.3 Reset Status.....</i>	145
<i>A.4 Digital Output .....</i>	146
<i>A.5 Latched Digital Input .....</i>	147

# 1. Introduction

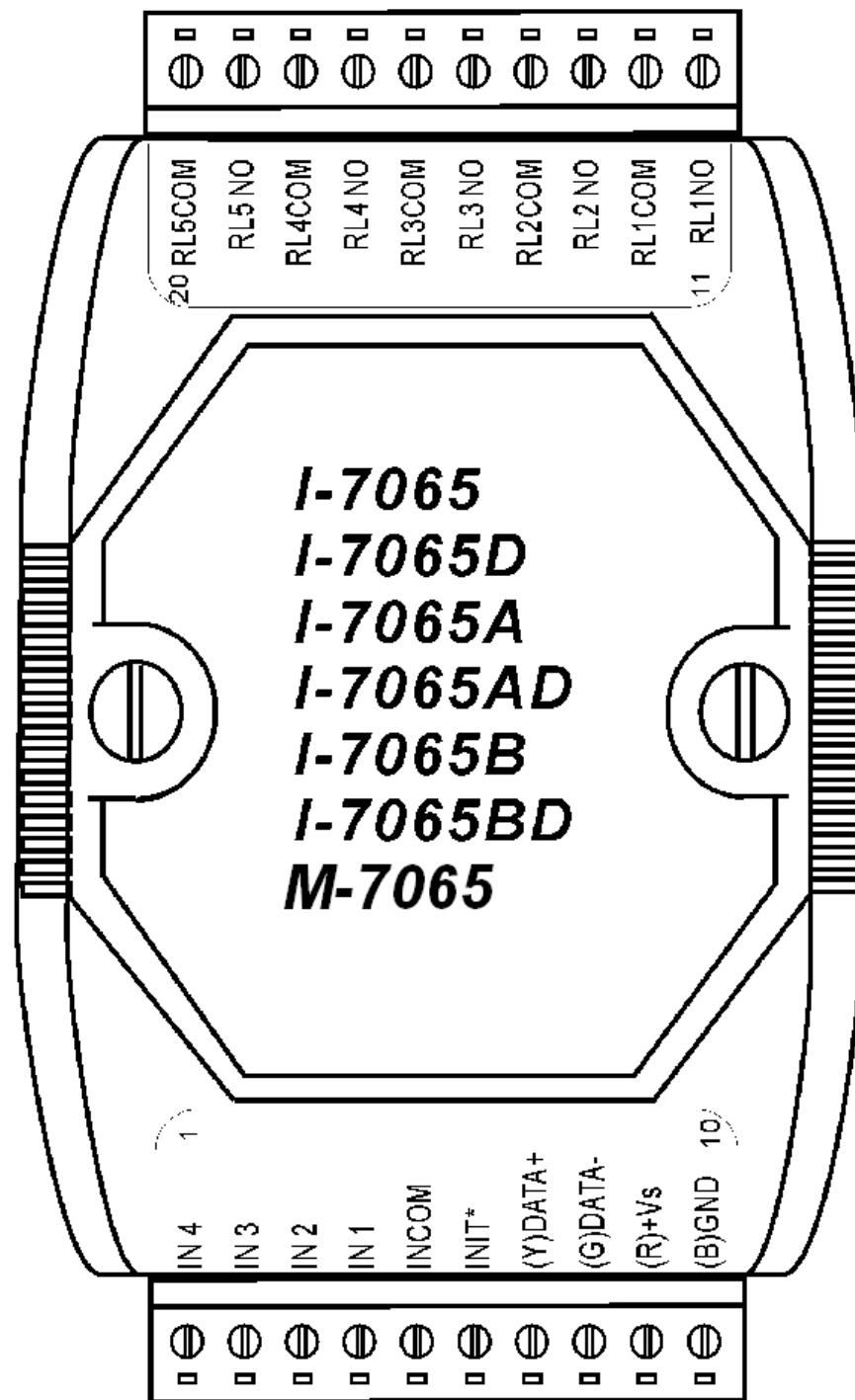
The I-7000 series is a family of network data acquisition and control modules, providing analog-to-digital, digital-to-analog, digital input/output, timer/counter and other functions. The modules can be remotely controlled using a set of commands, which we call the DCON protocol. Communication between the module and the host is in ASCII format via an RS-485 bi- directional serial bus standard. Baud Rates are software programmable and transmission speeds of up to 115.2 Kbps can be selected. The functionality of the M-7000 series is the same as the I-7000 series, with the exception that the M-7000 series offers extended support for the Modbus RTU protocol.

The I-7000 and M-7000 DIO modules support TTL signal, photo-isolated digital input, AC voltage digital input, relay contact output, solid-state relay output, photoMOS output and open-collector output.

## **1.1 More Information**

For more information regarding the I-7000 series, please refer to chapter 1 of the “I-7000 Bus Converter User’s Manual” as shown below or visit the ICP DAS website <http://www.icpdas.com>.

## 1.2 Pin Assignment

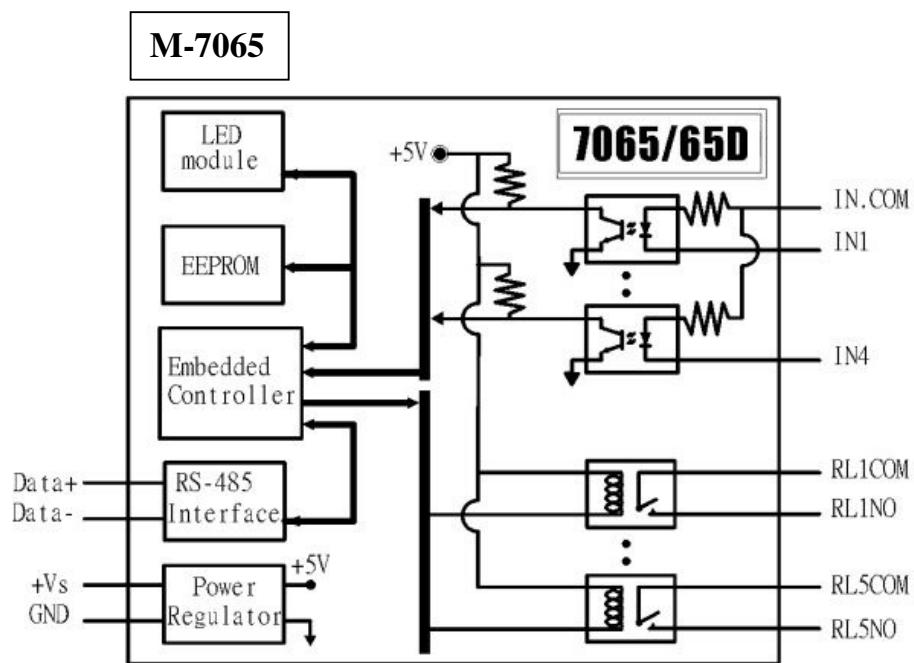


## 1.3 Specifications

<b>Digital Input</b>	
Channels	4
Input Type	Sink, Source, isolated channels with common ground or power
Off Voltage Level	+1V max.
On Voltage Level	+4 to +30V
Input Impedance	3K Ohms, 0.5W
Counters	channels : 4
	Max. Counters : 16-bit (65535)
	Max. Input Frequency : 100Hz
	Min. Pulse Width : 5ms
Photo-Isolation	3750Vrms
<b>Relay Output</b>	
Channels	5
Relay Type	Form "A" relay SPST (N.O.)
Contact Rating	AC: 250V @5A
	DC: 30V @5A
Operating Time(Max.)	6ms
Release Time(Min.)	3ms
Surge Strength	4,000VAC
Insulation Resistance	1000MOhms min. at 500VDC
Life Time	Mechanical : 20 X 10^6 OPS
	Electrical : 100 X 10^3 Full Load
<b>Interface</b>	
Interface	RS-485
Format	N, 8, 1 / N, 8, 2 / E 8, 1 / O 8, 1
Baud Rate	1200 ~ 115200bps
<b>LED Display</b>	
1 LED as Power/ Communication indicator	
4 LEDs as Digital Input indicators and 5 LEDs as Relay Output indicators (for I-7065D)	
<b>Power</b>	
Input Voltage Range	10~30VDC

Power Consumption	1.3W (max.)(I-7065) / 1.3W (max.)(I-7065D)
<b>Environment</b>	
Operating Temperature	-25 to 75 °C
Storage Temperature	-40 to 85 °C
Humidity	5 to 95%, non-condensing
<b>Ordering information</b>	
M-7065	4-channel Isolated Digital Input and 5-channel Relay Output Module with 16-bit Counters

## 1.4 Block Diagram



## 1.5 Wire Connection

	Dry Contact	
Input Type	A circuit diagram showing a dry contact input. The top wire is labeled '+' and the bottom wire is labeled '-'. They both connect to two parallel contacts in series with the IN.COM and INx pins.	A circuit diagram showing a dry contact input. The top wire is labeled '-' and the bottom wire is labeled '+'. They both connect to two parallel contacts in series with the IN.COM and INx pins.
	TTL/CMOS	
	A circuit diagram showing a TTL/CMOS input. A logic HIGH signal is connected to the IN.COM pin through a diode. A logic LOW signal is connected to the INx pin through a diode.	A circuit diagram showing a TTL/CMOS input. A logic LOW signal is connected to the IN.COM pin through a diode. A logic HIGH signal is connected to the INx pin through a diode.
Output Type	NPN Output	PNP Output
	A circuit diagram showing an NPN output. A base signal is connected to the IN.COM pin through a diode. The collector is connected to the Load and AC/DC power source. The emitter is connected to ground.	A circuit diagram showing a PNP output. A base signal is connected to the INx pin through a diode. The collector is connected to the Load and AC/DC power source. The emitter is connected to ground.
	Relay Collector	
	Relay ON	Relay OFF
	A circuit diagram showing a relay collector for Relay ON. The coil is connected between the AC/DC power source and the Load. The common terminal of the relay is connected to the RLx NO pin and the RLx COM pin.	A circuit diagram showing a relay collector for Relay OFF. The coil is connected between the AC/DC power source and the Load. The common terminal of the relay is connected to the RLx NO pin and the RLx COM pin.

## 1.6 Quick Start

This Quick Start document describes the methods used to quickly set up and test the M-7065 using the ICP DAS DCON Utility. After confirming that the module is operational, more applications or software support be found by referring to the following links:

For Win32:

[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_utility/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/)

For Linux:

[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_linux/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_linux/)

SDK tool:

Active X: [ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_activex/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_activex/)

DDE server: [ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_dde/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_dde/)

DLL: [ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_dll/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_dll/)

Indusoft: [ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_indusoft/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_indusoft/)

Labview: [ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_labview/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_labview/)

## Loading the DCON Utility

1. The required installation file can be downloaded from:

[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_utility/setup/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/setup/)

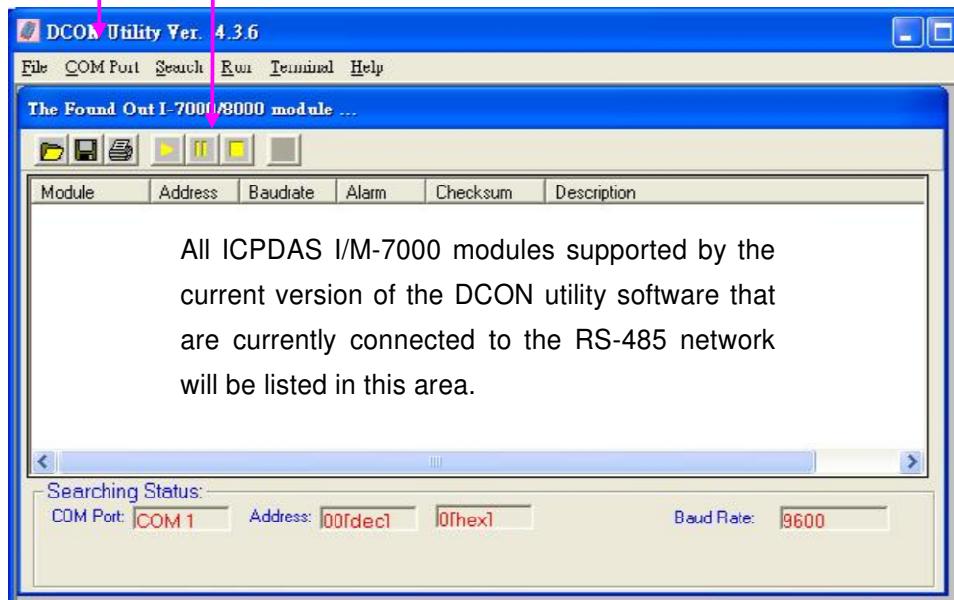
More information related to about version numbers and the latest supported modules can be obtained from

[ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon\\_utility/](ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/driver/dcon_utility/)

## Setting up and testing the module

1. After installing the DCON Utility, there will be a shortcut icon for the DCON Utility located on the Windows desktop. Double click this icon to launch the DCON Utility to set the module's configuration to fit with your application. After double clicking the icon, the following screen will be shown:

The tool bars are used to set the COM port environment, search for connected modules, data logs, etc.



- **How do I set the configuration of the modules ?**

ICP DAS M-7065 DIO module are command based. A series of commands are provided to allow the configuration and DI/O functions to be set. The basic DI/O and configuration commands are listed below. Refer to the following links for more information.

<ftp://ftp.icpdas.com/pub/cd/8000cd/napdos/7000/manual/modbusdio.pdf>

The Configuration command structure of the Modbus RTU is as follows:

Field ❶	Field ❷	Field ❸	Field ❹ ~ Field *n	Field (❺+*n)
Module Address	Function code	Sub function	Configuration field	CRC16

\*n: This value depends on the Sub-function code. Please refer to the “Modbus DIO User’s Manual” for more details.

Ex: To modify the power-on value of the module from 01, the following command should be sent:

01 46 27 0F BB F9

The supported DI/O commands are as follows:

Function code	Description
0x01	Read coils
0x02	Read discrete inputs
0x03	Read multiple registers
0x04	Read multiple input registers
0x05	Write single coils
0x0F	Write multiple coils

Ex: To read the current DI value of channels 0 to 5, the following command should be sent:

01 02 00 00 00 05 B8 09

Ex: To write the DO value 0x0F from channels 0 to 4, the following command should be sent:

01 0F 00 00 00 04 01 FF 7E D6

Ex: To only set the DO value of channel 2 to 1, the following command should be sent:

01 05 00 02 FF 00 2D FA

The Configuration command structure of the Modbus ASCII is as follows:

**Command Format:**

Leading Character	Module Address	Command	[LRC]	CR	LF
-------------------	----------------	---------	-------	----	----

**Response Format:**

Leading Character	Module Address	Data	[LRC]	CR	LF
-------------------	----------------	------	-------	----	----

Using Modbus ASCII Protocol, all command are coded in hexadecimal values, represented with readable ASCII characters. Only the characters 0...9 and A...F are used for coding.

Using Modbus ASCII Protocol, characters are used to start and end a frame. The Leading Character ':' is used to flag the start of a command and each command is ended with a CR•LF combination. The LRC characters are appended to the command preceding the CR•LF characters.

### **LRC Calculation:**

All characters except for Leading Character (:) and delimiter (CR•LF) are added with a carry being discarded. Total value is converted to binary notation, is converted to 2's complements, then to hexadecimal figures, that is, LRC.

### **Example :**

Modbus RTU => 01 46 00 [ 12 60 ]

Modbus ASCII => :014600B9(CR•LF)

(1.) hexadecimal = 01h+46h+00h = 47h

(2.) 2's complement: = B9h (LRC)

### **● Technical Support**

If you have problems about using the M-7065 DIO module, please contact ICP DAS Product Support.

Email: [Service@icpdas.com](mailto:Service@icpdas.com)

## 1.7 Default Settings

Default settings for the M-7065 modules are as follows :

- Protocol : DCON
- Module Address : 01
- Baud Rate : 9600 bps,n,8,1
- Checksum : Disabled

## 1.8 Configuration Tables

### Baud Rate Setting (CC)

7	6	5	4	3	2	1	0
Parity	Baud Rate Code						

#### Parity (bit6~bit7)

Code	00	01	10	11
Parity	n,8,1	n,8,2	e,8,1	o,8,1

#### Baud Rate Code (bit0~bit5)

Code	03	04	05	06	07	08	09	0A
Baud Rate	1200	2400	4800	9600	19200	38400	57600	115200

### Type Code (TT)

For M-7065 DIO modules, the type code is fixed to 40

## Data Format Setting (FF)

7	6	5	4	3	2	1	0
CU	CS	reserved			reserved		

Key	Description
CS	Checksum Setting 0: Disable 1: Enable
CU	Count Update 0: The counter is updated when there is a falling edge in the input signal. 1: The counter is updated when there is a rising edge in the input signal.

**Note:** The reserved bits should be zero.

## 1.9 DI/O Data Format Table

The data format of the response of the \$AA4, \$AA6 and \$AALS commands is:

**(the First Data)(the Second Data)00.**

The data format of the response of the @AA commands is:

**(the First Data)(the Second Data).**

**Note:** both the First Data and the Second Data are in two hexadecimal digits format.

Module	The First Data		The Second Data	
M-7065	DO0-DO5	00~1F	DI0-DI3	00~0F

## 2. DCON Protocol

All communication with M-7000 modules consists of commands generated by the host and responses transmitted by the M-7000 modules. Each module has a unique ID number that is used for addressing purposes and is stored in non-volatile memory. The ID is 01 by default and can be changed using a user command. All commands to the modules contain the ID address, meaning that only the addressed module will respond. The only exception to this is commands #\*\*\*(Section 2.2) and ~\*\*\*(Section 2.26), which are sent to all modules, but in both of these cases, the modules do not reply to the command.

### Command Format:

Leading Character	Module Address	Command	[CHKSUM]	CR
-------------------	----------------	---------	----------	----

### Response Format:

Leading Character	Module Address	Data	[CHKSUM]	CR
-------------------	----------------	------	----------	----

**CHKSUM** 2-character checksum which is present when the checksum setting is enabled. See Sections 1.1 (Data Format Setting) for details.

**CR** End of command character, carriage return (0x0D)

## **Checksum Calculation:**

1. Calculate the ASCII code sum of all the characters in the command/response string except for the carriage return character (CR).
2. The checksum is equal to the sum masked by 0FFh.

### **Example:**

Command string: \$012(CR)

1. Sum of the string = “\$”+”0”+”1”+”2” =  
 $24h + 30h + 31h + 32h = B7h$
2. Therefore the checksum is B7h, and so  
CHKSUM = “B7”
3. The command string with the checksum = \$012B7(CR)

Response string: !01200600(CR)

1. Sum of the string =  
“!”+”0”+”1”+”2”+”0”+”0”+”6”+”0”+”0” =  
 $21h + 30h + 31h + 32h + 30h + 30h + 36h + 30h + 30h = 1AAh$
2. Therefore the checksum is AAh, and so  
CHKSUM = “AA”
3. The response string with the checksum  
= !01200600AA(CR)

Note:

All characters should be in upper case.

General Command Sets			
Command	Response	Description	Section
%AANNTTCCFF	!AA	Sets the Module Configuration	2.1
#**	No Response	Synchronized Sampling	2.2
#AA00(Data)	>	Sets the Digital Output	2.3
#AA0A(Data)	>	Sets the Digital Output	2.4
#AA1cDD	>	Sets the Digital Output	2.5
#AAAcDD	>	Sets the Digital Output	2.6
#AAN	!AA(Data)	Reads the Digital Input Counter	2.7
\$AA2	!AANNTTCCFF	Reads the Module Configuration	2.8
\$AA4	!S(Data)	Reads the Synchronized Data	2.9
\$AA5	!AAS	Reads the Reset Status	2.10
\$AA6	!(Data)	Reads the Digital I/O Status	2.11
\$AAC	!AA	Clears the Latched DI Status	2.12
\$AACN	!AA	Clears the Digital InputCounter	2.13
\$AAF	!AA(Data)	Reads the firmware Version	2.14
\$AALS	!(Data)	Reads the Latched DI Status	2.15
\$AAM	!AA(Data)	Reads the Module Name	2.16
\$AAP	!AASC	Reads the communication protocol	2.17
\$AAPN	!AA	Sets the communication protocol	2.18
@AA	>(Data)	Reads the Digital I/O Status	2.19
@AA(Data)	>	Sets the Digital Output Channels	2.20
~AAO(Name)	!AA	Sets the Module Name	2.21
~AAD	!AAF	Reads the DI/O active status.	2.22
~AADVV	!AA	Sets the DI/O active status.	2.23
~AAI	!AA	Sets the soft INIT	2.24
~AATnn	!AA	Set the soft INIT timeout value	2.25

Host Watchdog Command Sets			
Command	Response	Description	Section
~**	No Response	Host OK	2.26
~AA0	!AASS	Reads the Status	2.27
~AA1	!AA	Resets the Status	2.28
~AA2	!AAVV	Reads the Timeout Settings	2.29
~AA3EVV	!AA	Sets the Timeout Settings	2.30
~AA4V	!AA(Data)	Reads the PowerOn/Safe Value	2.31
~AA5V	!AA	Sets the PowerOn/Safe Value	2.32
~AARDvv	!AA	Sets the Response Delay Time	2.33
~AARD	!AA(Data)	Reads the Response Delay Time	2.34

## **2.1 %AANNTTCCFF**

### **Description:**

Sets the configuration of a module.

### **Syntax:**

**%AANNTTCCFF[CHKSUM](CR)**

<b>%</b>	Delimiter character
<b>AA</b>	Address of the module to be configured in hexadecimal format (00 to FF)
<b>NN</b>	New address of the module in hexadecimal format (00 to FF)
<b>TT</b>	Type code, should be 40 for DIO module.
<b>CC</b>	(Baud Rate code, should be 0A for DIO)
<b>FF</b>	Used to set the counter update direction and checksum (Section 1.1).

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

**!** Delimiter for a valid command.

**?** Delimiter for an invalid command.

**AA** Address of the module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:%0101400A80

Response:!01

Sets the counter update (Rising Edge)

FF (Sets the counter update)

80 : Sets the counter update (Rising Edge)

00 : Sets the counter update (Falling Edge)

Command:%0101400A00

Response:!01

Sets the counter update (Falling Edge)

FF (Sets the counter update)

80 : Sets the counter update (Rising Edge)

00 : Sets the counter update (Falling Edge)

### **Related Commands:**

Section 2.8 \$AA2 、 2.24 ~AAI 、 2.25 ~AATnn

### **Related Topics:**

Section 1.1 Configuration Tables

## **2.2 #\*\***

### **Description:**

When the command is received, it will allow all modules to read data and will store the data for later retrieval.

### **Syntax:**

**#\*\*[CHKSUM](CR)**

#            Delimiter character

\*\*          Synchronized sampling command

### **Response:**

There is no response with this command. To access the data, another command, \$AA4, must be sent, see Section 2.11 for details.

### **Examples:**

Command:#\*\*

Response: No response

Sends the synchronized sampling command.

Command:\$014

Response:!10C0300

Sends a command to read the synchronized data.  
The status byte of the response is 1, which means  
that it is the first time the synchronized data has  
been read after the previous #\*\* command.

Digital Output : 0x0C

Digital Input : 0x03

Command:\$014

Response:!00C0300

Sends a command to read the synchronized  
data. The status byte of the response is 0,  
which means that it is NOT the first time the  
synchronized data has been read after the  
previous #\*\* command.

Digital Output : 0x0C

Digital Input : 0x03

## **Related Commands:**

Section 2.9 \$AA4

## **2.3 #AA00(Data)**

### **Description:**

Sets the digital output value of the lower eight channels.  
(DO0-DO7)

### **Syntax:**

**#AA00(Data)[CHKSUM](CR)**

#            Delimiter character  
AA         Address of the module to be set (00 to FF)  
00         Command to set the digital output value of  
              the lower eight channels. (DO0-DO7)  
(Data)      A two-digit hexadecimal value, where bit 0  
              corresponds to DO0, bit 1 corresponds to  
              DO1, etc. When the bit is 1, it denotes that  
              the digital output channel is on, and 0  
              denotes that the digital output channel is off.

### **Response:**

Valid Command:    >[CHKSUM](CR)

Invalid Command: ?AA[CHKSUM](CR)

Ignored Command: ! [CHKSUM](CR)

>            Delimiter for a valid command.

?            Delimiter for an invalid command.

!            Ignored Delimiter character.

A host watchdog timeout has occurred, the digital output channel s are set to safe value, and the digital output value that was sent is ignored.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:#010006

Response:>

Sets DO0、DO3 to off , DO1、DO2 to on, and the module returns a valid response.

Command:#010016

Response:>

Sets DO0、DO3 to off , DO1、DO2 to on  
DO6、DO7、DO8 to off , DO5 to on

Command:#010005

Response:!

Host Watchdog Timeout.

## **Related Commands:**

Section 2.4 #AA0A(Data)、2.5 #AA1cDD  
2.6#AAAcDD、2.11 \$AA6、2.19 @AA  
2.20 @AA(Data)

## **2.4 #AA0A(Data)**

### **Description:**

Sets the digital output value.(DO0-DO31)

### **Syntax:**

**#AA0A(Data)[CHKSUM](CR)**

#	Delimiter character
AA	Address of the module to be set (00 to FF)
0A	Command to set the digital output value. (DO0-DO31)  #AA0A(Data) : DO0-DO7 #AA0B(Data) : DO8-DO15 #AA0C(Data) : DO16-DO23 #AA0D(Data) : DO24-DO31
(Data)	A two-digit hexadecimal value, where bit 0 corresponds to DO0, bit 1 corresponds to DO1, etc. When the bit is 1, it denotes that the digital output channel is on, and 0 denotes that the digital output channel is off.

## **Response:**

Valid Command: >[CHKSUM](CR)

Invalid Command: ?AA[CHKSUM](CR)

Ignored Command: !(CHKSUM)(CR)

> Delimiter for a valid command.

? Delimiter for an invalid command.

! Ignored Delimiter character

watchdog timeout has occurred, the digital output channel s are set to safe value, and the digital output value that was sent is ignored.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:#010A0E

Response:>

Sets DO0 to off , DO1 、 DO2 、 DO3 to on.

Command:#010A06

Response:!

Host Watchdog Timeout.

Command:#010A0F

Response:>

Sets DO0、DO1、DO2、DO3 to on.

Command:#020A00

Response:>

Sets DO0、DO1、DO2、DO3 to off.

### **Related Commands:**

Section 2.3 #AA00(Data)、2.5 #AA1cDD

2.6#AAAcDD、2.11 \$AA6、2.19 @AA

2.20 @AA(Data)

## **2.5 #AA1cDD**

### **Description:**

Sets a single digital output channel of the lower eight channels.

### **Syntax:**

**#AA1cDD [CHKSUM](CR)**

#        Delimiter character  
AA      Address of the module to be set (00 to FF)  
1        Command to set a single digital output  
          channel of the lower eight channels.  
c        Specifies the digital output channel to be set  
          (0 to 7).  
DD      00 : set the digital output channel to off.  
          01 : set the digital output channel to on.

### **Response:**

Valid Command:    >[CHKSUM](CR)

Invalid Command: ?AA[CHKSUM](CR)

Ignored Command: ! [CHKSUM](CR)

>        Delimiter for a valid command.

?        Delimiter for an invalid command.

!        Ignored Delimiter character

watchdog timeout has occurred, the digital output channel s are set to safe value, and the digital output value that was sent is ignored.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:#021001

Response:>

Sets DO0 to on.

Command:#021401

Response:>

Command:#021301

Response:>

Sets DO3 to on.

### **Related Commands:**

Section 2.3 #AA00(Data) 、 2.4 #AA0A(Data)

2.6#AAcDD 、 2.11 \$AA6 、 2.19 @AA

2.20 @AA(Data)

## **2.6 #AAAcDD**

### **Description:**

Sets a single digital output channel. (DO0-DO31)

### **Syntax:**

**#AAAcDD [CHKSUM](CR)**

#	Delimiter character
AA	Address of the module to be set (00 to FF)
A	Command to set a single digital output channel. (DO0-DO31) #AAAcDD : DO0-DO7 #AABcDD : DO8-DO15 #AACcDD : DO16-DO23 #AADcDD : DO24-DO31
c	Specifies the digital output channel to be set. (0-7)
DD	00 : set the digital output channel to off. 01 : set the digital output channel to on.

### **Response:**

Valid Command:   **>[CHKSUM](CR)**

Invalid Command:   **?AA[CHKSUM](CR)**

Ignored Command:   **![CHKSUM](CR)**

- > Delimiter for a valid command.
- ? Delimiter for an invalid command.
- ! Ignored Delimiter character
  - watchdog timeout has occurred, the digital output channel s are set to safe value, and the digital output value that was sent is ignored.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:#02A201

Response:>

Sets DO2 to on

### **Related Commands:**

Section 2.3 #AA00(Data) 、 2.4 #AA0A(Data)  
2.5 #AA1cDD 、 2.11 \$AA6 、 2.19 @AA  
2.20 @AA(Data)

## **2.7 #AAN**

### **Description:**

Reads the digital input counter of channel N.

### **Syntax:**

**#AAN [CHKSUM](CR)**

#            Delimiter character  
AA         Address of the module to be set (00 to FF)  
N         The channel to be read (0 to F)

### **Response:**

Valid Command:   !(Data)[CHKSUM](CR)

Invalid Command: ?AA[CHKSUM](CR)

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
An invalid command is returned if the specified channel is incorrect.  
(Data)     Five decimal digits representing the digital input counter data of the specified channel (00000 to 65535).  
AA         Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:#011

Response:!0100005

Reads data from channel 1 of module 01 and the returned counter value is 00005.

Command:#015

Response:!0100005

Reads data from channel 5 of module 01 and the returned counter value is 00005.

**Related Commands:**

Section 2.13 #AACN

## **2.8 \$AA2**

### **Description:**

Reads the module configuration.

### **Syntax:**

**\$AAN [CHKSUM](CR)**

\$            Delimiter character  
AA         Address of the module to be set (00 to FF)  
2            Command to read the module configuration

### **Response:**

Valid Command:   **!AATTCCFF[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA         Address of the responding module (00 to FF)  
TT         Type code of the module, should be 40 for  
            DIO module.  
CC         Baud Rate code of the module, should be 0A  
FF         Checksum and counter update direction  
            settings of the module, see Section 1.1 for  
            details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:\$012

Response:!01400A80

Reads the configuration of module 01.

FF (Sets the counter update)

80 : Reads the counter update (Rising Edge)

00 : Reads the counter update (Falling Edge)

Command:\$012

Response:!01400AC0

Reads the configuration of module 01.

FF (Sets the counter update)

C0 : Reads the counter update (Rising Edge),  
checksum enabled.

**Related Commands:**

Section 2.1 %AANNTTCCFF

**Related Topics:**

Section 1.1

## **2.9 \$AA4**

### **Description:**

Reads the synchronized data that was retrieved by the last #\*\* command.

### **Syntax:**

**\$AA4 [CHKSUM](CR)**

\$            Delimiter character  
AA         Address of the module to be set (00 to FF)  
4            Command to read the synchronized data

### **Response:**

Valid        !S[CHKSUM](CR)

Command:

Invalid      ?AA[CHKSUM](CR)

Command:

!            Delimiter for a valid command.

?            Delimiter for an invalid command.

AA         Address of the responding module (00 to FF)

S            Status of the synchronized data

1 : first read

0 : not the first read

(Data)      Synchronized data. See Section 1.2 for the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:#\*\*

Response: No response

Sends the synchronized sampling command.

Command:\$014

Response:!10F0000

Reads the synchronized data for module 01.

The module returns the synchronized data and sets the status byte to 1 to indicate that this is the first time the synchronized data has been read. Digital Output : 0x0F

Command:\$014

Response:!0053F00

Reads the synchronized data for module 01.

The module returns the synchronized data and sets the status byte to 0 to indicate that the synchronized data has been read

Digital Output : 0x05

Digital Input : 0x3F

**Related Commands:**

Section 2.2 #\*\*

## **2.10 \$AA5**

### **Description:**

Reads the reset status of a module.

### **Syntax:**

**\$AA5 [CHKSUM](CR)**

\$            Delimiter character  
AA          Address of the module to be set (00 to FF)  
5            Command to read the module reset status.

### **Response:**

Valid Command:   **!AAS[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA          Address of the responding module (00 to FF)  
S            Reset status of the module  
    1 : This is the first time the command has  
          been sent since the module was powered on.  
    0 : This is not the first time the command  
          has been sent since the module was  
          powered on, which denotes that there has  
          been no module reset since the last \$AA5  
          command was sent.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:\$015

Response:!011

Reads the reset status of module 01. The response shows that it is the first time the \$AA5 command has been sent since the module was powered-on.

Command:\$015

Response:!010

Reads the reset status of module 01. The response shows that there has been no module reset since the last \$AA5 command was sent.

## **2.11 \$AA6**

### **Description:**

Reads the status of the digital input/output channels.

### **Syntax:**

**\$AA6 [CHKSUM](CR)**

\$            Delimiter character  
AA          Address of the module to be set (00 to FF)  
6            Command to read the digital input/output  
              channels

### **Response:**

Valid Command:   **!(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA          Address of the responding module (00 to FF)  
(Data)     Status of the digital input/output channels, a  
              four- digit hexadecimal value followed by  
              00. See Section 1.2 for details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:\$016

Response:!1F0000

Reads the digital input/output channel status of module 01. Digital Output : 0x1F

Command:\$016

Response:!070F00

Reads the digital input/output channel status of module 01.

Digital Output : 0x07

Digital Input : 0x0F

## **Related Commands:**

Section 2.3 #AA00(Data) 、 2.4 #AA0A(Data)

2.5 #AA1cDD 、 2.6 #AAAcDD 、 2.19 @AA

2.20 @AA(Data)

## **2.12 \$AAC**

### **Description:**

Clears the status of the latched digital input channels.

### **Syntax:**

**\$AAC [CHKSUM](CR)**

\$            Delimiter character  
AA         Address of the module to be set (00 to FF)  
C         Command to clear the status of the latched  
            digital input channels

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA         Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:@016

Response:>

Sets DO1、2 to on、DO0、3 to off

Command:\$01C

Response:!01

Sends the command to clear the status of the latched digital input channels of module 01 and returns a valid response.

Command:\$01L0

Response:!093F00

Sends the command to read the status of the low latched digital input channels of module 01.

Low Latched DI : 0x3F

Low Latched DO : 0x09

## **Related Commands:**

Section 2.15 \$AALS

## **2.13 \$AACN**

### **Description:**

Clears the digital input counter of channel N.

### **Syntax:**

**\$AACN [CHKSUM](CR)**

\$            Delimiter character  
AA         Address of the module to be set (00 to FF)  
C         Command to clear the digital input counter  
N         The channel to be cleared (0 to F).

### **Response:**

Valid Command:   **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA         Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:#011

Response:!0100009

Reads counter data from channel 1 of module 01  
and the returned counter value is 00009.

Command:\$01C1

Response:!01

Clears the counter value of channel 1 of module  
01 and returns a valid response.

Command:#011

Response:!0100000

Reads counter data from channel 1 of module  
01 and the returned counter value is 0.

## **Related Commands:**

Section 2.7 #AAN

## **2.14 \$AAF**

### **Description:**

Reads the firmware version of a module.

### **Syntax:**

**\$AAF [CHKSUM](CR)**

\$            Delimiter character  
AA         Address of the module to be set (00 to FF)  
F            Command to read the firmware version

### **Response:**

Valid Command:   **!AA(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA         Address of the responding module (00 to FF)  
(Data)     Firmware version string of the module

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:\$01F

Response:!0102.00

Reads the firmware version of module 01, and shows that it is version 02.00.

## **2.15 \$AALS**

### **Description:**

Reads the status of the latched digital input channels.

### **Syntax:**

**\$AALS [CHKSUM](CR)**

\$            Delimiter character  
AA          Address of the module to be set (00 to FF)  
L            Command to read the latched status  
S            0 : read the low latched status  
              1 : read the high latched status

### **Response:**

Valid Command:    **!(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA          Address of the responding module (00 to FF)  
(Data)      Status of the latched digital input  
              channels, a four- digit hexadecimal value  
              followed by 00. See Section 1.2 for  
              details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:@016

Response:>

Sets DO1、2 to on、DO0、3 to off.

Command:\$01C

Response:!01

Sends the command to clear the status of the latched digital input channels of module 01 and returns a valid response.

Command:\$01L0

Response:!093F00

Sends the command to read the status of the low latched digital input channels of module 01.

Low Latched DI : 0x3F

Low Latched DO : 0x09

Command:\$01L1

Response:!060000

Sends the command to read the status of the high latched digital input channels of module 01.

High Latched DI : 0x00

High Latched DO : 0x06

### **Related Commands:**

Section 2.12 \$AAC

### **Related Topics:**

Section 1.2

## **2.16 \$AAM**

### **Description:**

Reads the name of a module.

### **Syntax:**

**\$AAM [CHKSUM](CR)**

\$            Delimiter character  
AA         Address of the module to be set (00 to FF)  
M         Command to read the module name

### **Response:**

Valid Command:   **!AA(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA         Address of the responding module (00 to FF)  
(Data)     Name string of the module

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Command:\$01M**

**Response:!017065**

Reads the module name of module 01 and returns the name.

**Related Commands:**

Section 2.21 ~AAO(Name)

## **2.17 \$AAP**

### **Description:**

Reads the communication protocol information.

### **Syntax:**

**\$AAP [CHKSUM](CR)**

\$            Delimiter character  
AA         Address of the module to be set (00 to FF)  
P            Command to read the communication  
              protocol

### **Response:**

Valid Command:   **!AASC[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA         Address of the responding module (00 to FF)  
S            The protocols supported by the module  
              0: DCON  
              1: DCON and Modbus RTU  
              3: DCON and Modbus RTU/ASCII  
C            The current protocol that is saved in the  
              EEPROM that will be used at the next  
              power-on reset

- 0: DCON
- 1: Modbus RTU
- 3: Modbus ASCII

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:\$01P

Response:!0130

Reads the communication protocol of module 01 and returns a response of 30 meaning that it supports the DCON and Modbus RTU/ASCII protocols and the protocol that will be used at the next power-on reset is DCON.

### **Related Commands:**

Section 2.18 \$AAPN

## **2.18 \$AAPN**

### **Description:**

Sets the communication protocol.

### **Syntax:**

**\$AAP [CHKSUM](CR)**

\$            Delimiter character  
AA        Address of the module to be set (00 to FF)  
P        Command to set the communication protocol  
N        0: DCON  
            1: Modbus RTU  
            3: Modbus ASCII

### **Response:**

Valid Command:   **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!            Delimiter for a valid command.  
?            Delimiter for an invalid command.  
AA        Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:\$01P1

Response:?01

Sets the communication protocol of module 01 to Modbus RTU and returns an invalid response because the module is not in INIT mode.

Command:\$01P1

Response:!01

Sets the communication protocol of module 01 to Modbus RTU and returns a valid response.

### **Related Commands:**

Section 2.17 \$AAP

## **2.19 @AA**

### **Description:**

Reads the status of the digital input/output ports.

### **Syntax:**

**@AA [CHKSUM](CR)**

**@** Delimiter character

**AA** Address of the module to be set (00 to FF)

### **Response:**

Valid Command: **>(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

**>** Delimiter for a valid command.

**?** Delimiter for an invalid command.

**AA** Address of the responding module (00 to FF)

**(Data)** Status of the digital input/output ports, a four-digit hexadecimal value. See Section 1.2 for details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:@02

Response:>0F0F

Reads the digital input/output port status of module 02 and returns >030F, which denotes that RL1, RL2, RL3 and RL4 are on and IN1, IN2, IN3 and IN4 are on .

## **Related Commands:**

Section 2.3 #AA00(Data) 、 2.4 #AA0A(Data)

2.5 #AA1cDD 、 2.6 #AAAcDD 、 2.11 \$AA6

2.20 @AA(Data)

## **Related Topics:**

Section 1.2

## **2.20 @AA(Data)**

### **Description:**

Sets the digital output channels.

### **Syntax:**

**@AA(Data)[CHKSUM](CR)**

**@** Delimiter character

**AA** Address of the module to be set (00 to FF)

**(Data)** Data to be written to the digital output channels.

Bit 0 of the value corresponds to DO0 and bit 1 of the value corresponds to DO1, etc.

When the bit is 1, it denotes that the digital output channel is on, and 0 denotes that the digital output channel is off.

### **Response:**

Valid Command: **>[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

Ignored Command: **![CHKSUM](CR)**

**>** Delimiter for a valid command.

**?** Delimiter for an invalid command.

**!** Ignored Delimiter character

watchdog timeout has occurred, the digital output ports are set to safe value, and the digital output value that was sent is ignored.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:@02F

Response:>

Sets DO0、DO1、DO2、DO3 to on.

### **Related Commands:**

Section 2.3 #AA00(Data)、2.4 #AA0A(Data)、  
2.5 #AA1cDD、2.6#AAAcDD、2.11  
\$AA6、2.19 @AA

## **2.21 ~AAO(Name)**

### **Description:**

Sets the name of a module.

### **Syntax:**

**~AAO(Name) [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
O Command to set the module name  
**(Name)** New name of the module (max. 6 characters)

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:\$01M

Response:!017065

Reads the name of module 01 and returns the name 7065.

Command:~01OM7065

Response:!01

Sets the name of module 01 to be “ M7065” and returns a valid response.

Command:\$01M

Response:!01M7065

Reads the name of module 01 and returns the name M7065.

## **2.22 ~AAD**

### **Description:**

Reads Digital I/O Active status.

### **Syntax:**

**~AAD [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
D Command to read the DI/O active status

### **Response:**

Valid Command: **!AAVV[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module (00 to FF)  
VV A two-digit hexadecimal value indicating the DI/O active status.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:~02D03

Response:!02

Sets DI/O active status.

Command:~02D

Response:!0203

Reads DI/O active status.

7	6	5	4	3	2	1	0
Reserved						OAS	IAS

Key	Description
OAS	DO Active Status 0: output value 1 for relay active output value 0 for relay inactive 1: output value 0 for relay active output value 1 for relay inactive
IAS	DI Active Status 0: input value 1 for non-signal or the low voltage; input value 0 for high voltage 1: input value 0 for non-signal or the low voltage; input value 1 for high voltage

(DI/O Active Status)

## **Related Commands:**

Section 2.23 ~AADVV

## **2.23 ~AADVV**

### **Description:**

Sets Digital I/O Active status.

### **Syntax:**

**~AADVV [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
D Command to set the DI/O active status.  
VV A two-digit hexadecimal value indicating the DI/O active status.

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:~02D03

Response:!02

Sets DI/O active status.

Command:~02D

Response:!0203

Reads DI/O active status.

7	6	5	4	3	2	1	0
Reserved						OAS	IAS

Key	Description
OAS	DO Active Status 0: output value 1 for relay active output value 0 for relay inactive 1: output value 0 for relay active output value 1 for relay inactive
IAS	DI Active Status 0: input value 1 for non-signal or the low voltage; input value 0 for high voltage 1: input value 0 for non-signal or the low voltage; input value 1 for high voltage

## **Related Commands:**

Section 2.22 ~AAD

## **2.24 ~AAI**

### **Description:**

The Soft INIT command is used to enable modification of the Baud Rate and checksum settings using software only.

### **Syntax:**

**~AAI [CHKSUM](CR)**

**~** Delimiter character

**AA** Address of the module to be set (00 to FF)

**I** Command to set the Sort INIT

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

**!** Delimiter for a valid command.

**?** Delimiter for an invalid command.

**AA** Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:~03I

Response:!03

Sets Soft INIT

Command:%0303400700

Response:?03

Attempts to change the Baud Rate of module 03 to 19200 without first adjusting the INIT \*pin. The module returns an invalid response because the soft INIT timeout value is 0.

Command:~03T10

Response:!03

Set the soft INIT time out value of module 01 to 16 seconds and returns a valid response.

Command:~03I

Response:!03

Set the Soft INIT of module 01 and return a valid response.

**Command:**%0301400740

**Response:**!03

Attempts to change the Baud Rate of module 01 to 19200 without first adjusting the INIT \* pin.The module returns an invalid response.

**Related Commands :**

Section 2.25 ~AATnn

## **2.25 ~AATnn**

### **Description:**

Sets the soft INIT timeout value.

### **Syntax:**

**~AATnn [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
T Command to set the Sort INIT timeout  
value  
nn Two hexadecimal digits representing the  
time out value in seconds. The maximum  
timeout value is 60 seconds.

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **2.26 ~\*\***

### **Description:**

Informs all modules that the host is OK.

### **Syntax:**

**~\*\* [CHKSUM](CR)**

~ Delimiter character

\*\* Host OK command

### **Response:**

No response

### **Examples:**

Command: ~\*\*

Response: No response

### **Related Commands:**

Section 2.27 ~AA0、2.28 ~AA1、2.29 ~AA2  
2.30 ~AA3EVV、2.31 ~AA4V、  
2.32 ~AA5V

## **2.27 ~AA0**

### **Description:**

Reads the host watchdog status of a module.

### **Syntax:**

**~AA0 [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
0 Command to read the module status

### **Response:**

Valid Command: **!AASS[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module (00 to FF)  
SS Two hexadecimal digits that represent the host watchdog status, where:  
Bit 7: 0 indicates that the host watchdog is disabled, and 1 indicates that the host watchdog is enabled,  
Bit 2: 1 indicates that a host watchdog timeout has occurred, and 0 indicates that no host watchdog timeout has occurred.

The host watchdog status is stored in EEPROM and can only be reset by using the ~AA1 command.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:~030

Response:!0304

Reads the host watchdog status of module 03 and returns 04, meaning that a host watchdog timeout has occurred.

Command:~030

Response:!0300

Reads the host watchdog status of module 03 and returns 00, meaning that the host watchdog is disabled and no host watchdog timeout has occurred.

Command:~030

Response:!0380

Reads the host watchdog status of module 03 and returns 08, meaning that the host watchdog is enabled. Sends ~\*\* to clear watchdog timeout value.

### **Related Commands:**

Section 2.26 ~\*\* 、 2.28 ~AA1 、 2.29 ~AA2  
2.30 ~AA3EVV 、 2.31 ~AA4V 、  
2.32 ~AA5V

## **2.28 ~AA1**

### **Description:**

Resets the host watchdog timeout status of a module.

### **Syntax:**

**~AA1 [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
1 Command to reset the host watchdog timeout  
status

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:~030

Response:!0304

Reads the host watchdog status of module 03 and shows that a host watchdog timeout has occurred.

Command:~031

Response:!03

Resets the host watchdog timeout status of module 03 and returns a valid response.

## **Related Commands:**

Section 2.26 ~\*\*、2.27 ~AA0、2.29 ~AA2

2.30 ~AA3EVV、2.31 ~AA4V、

2.32 ~AA5V

## **2.29 ~AA2**

### **Description:**

Reads the host watchdog timeout value of a module.

### **Syntax:**

**~AA0 [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
2 Command to read the host watchdog timeout  
value

### **Response:**

Valid Command: **!AAEVV[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module (00 to FF)  
E 1: the host watchdog is enabled  
0: the host watchdog is disabled  
VV Two hexadecimal digits to represent the timeout value in tenths of a second, for example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:`~032`

Response:`!03190`

Reads the host watchdog timeout value of module 03 and returns 0x90, which denotes that the host watchdog is enabled and the host watchdog timeout value is 14.4 seconds

### **Related Commands:**

Section 2.26 `~**`、2.27 `~AA0`、2.28 `~AA1`  
`2.30 ~AA3EVV`、`2.31 ~AA4V`、  
`2.32 ~AA5V`

## **2.30 ~AA3EVV**

### **Description:**

Enables/disables the host watchdog and sets the host watchdog timeout value of a module.

### **Syntax:**

**~AA3EVV [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
3 Command to set the host watchdog  
E 1: enable the host watchdog  
    0: disable the host watchdog  
VV Two hexadecimal digits to represent the timeout value in tenths of a second, for example, 01 denotes 0.1 seconds and FF denotes 25.5 seconds.

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

### **Examples:**

Command:`~013164`

Response:`!01`

Enables the host watchdog of module 01 and sets the host watchdog timeout value to 10.0 seconds.  
The module returns a valid response.

Command:`~012`

Response:`!01164`

Reads the host watchdog timeout value of module 01. The module returns 164, which denotes that the host watchdog is enabled and the host watchdog timeout value is 10.0 seconds.

### **Related Commands:**

Section 2.26 `~**`、2.27 `~AA0`、2.28 `~AA1`  
`2.29 ~AA2`、`2.31 ~AA4V`、  
`2.32 ~AA5V`

## **2.31 ~AA4V**

### **Description:**

Reads the power-on DO value or the safe DO value of a module.

### **Syntax:**

**~AA4V [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
4 Command to read the power-on DO value or  
the safe DO value  
V P: Power On Value  
S: Safe Value

### **Response:**

Valid Command: **!AA(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module(00 to FF)  
(Data) Power On Value or Safe Value

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:~034P

Response:!030700

Reads the power-on DO value of module 03 and returns 0700.

Command:~034S

Response:!030F00

Reads the safe DO value of module 03 and returns 0F00.

**Related Commands:**

Section 2.32 ~AA5V

## **2.32 ~AA5V**

### **Description:**

Sets the current DO value as the power-on DO value or the safe DO value.

### **Syntax:**

**~AA5V [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
5 Command to set the power-on value or the safe DO value  
V P: Power On Value  
S: Safe Value

### **Response:**

Valid Command: **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:@037

Response:>

Sets DO0、DO1、DO2 to on , DO3 to off

Command:~035P

Response:!03

Sets the power-on DO value and the module  
returns a valid response.

Command:@03F

Response:>

Sets DO0、DO1、DO2、DO3 to on

Command:~035S

Response:!03

Sets the safe DO value and the module returns  
a valid response.

Command:~034S

Response:!030F00

Reads the safe DO value of module 03.

## **Related Commands:**

Section 2.31 ~AA4V

## **2.33 ~AARDvv**

### **Description:**

Sets the Response Delay Time of a module.

### **Syntax:**

**~AARDvv [CHKSUM](CR)**

\$      Delimiter character

AA      Address of the module to be set (00 to FF)

RD      Command to set the Response Delay Time  
of a module.

vv      Response Delay Time (0~30,in 1ms)

### **Response:**

Valid Command:   **!AA[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!      Delimiter for a valid command.

?      Delimiter for an invalid command.

AA      Address of the responding module (00 to  
FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

## **Examples:**

Command:~03RD1E

Response:!03

Sets the Response Delay Time, the value is 1E.  
(30ms)

Command:~03RD

Response:!0300

Reads the Response Delay Time, the value is 00.  
(0ms).

## **Related Commands:**

Section 2.34 ~AARD

## **2.34 ~AARD**

### **Description:**

Reads the Response Delay Time of a module.

### **Syntax:**

**~AARD [CHKSUM](CR)**

~ Delimiter character  
AA Address of the module to be set (00 to FF)  
RD Command to read the Response Delay Time of  
a module.(0~30,in 1ms)

### **Response:**

Valid Command: **!AA(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter for a valid command.  
? Delimiter for an invalid command.  
AA Address of the responding module(00 to FF)  
(Data) Response Delay Time Value

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

**Examples:**

Command:~03RD1E

Response:!03

Sets the Response Delay Time, the value is 1E.  
(30ms)

Command:~03RD

Response:!0300

Reads the Response Delay Time, the value is 00.  
(0ms).

**Related Commands:**

Section 2.33 ~AARDvv

### 3. Modbus RTU Protocol

The Modbus protocol is developed by Modicon Inc., originally developed for Modicon controllers. Detailed information can be found at <http://www.modicon.com/techpubs/toc7.html>. You can also visit <http://www.modbus.org> to find more valuable information.

Address Mapping		
Address	Description	Attribute
00001	Digital Output Channel	R/W
00033	Digital Input Channel	R
10001	Digital Input Channel	R
00065	High Latched DI/O Channels	R
00097	Low Latched DI/O Channels	R
00129	Safe Value	R/W
00161	Power On Value	R/W
40485	Module address, valid range:1~247	R/W
40486	bits 5:0 (Baud rate) Baud rate, valid range:0x03-0x0A bits 7:6 00: no parity, 1 stop bit 10: even parity, 1 stop bit 11: odd parity, 1 stop bit	R/W
00264	Clear Latched DI/O	W
00513	Clear DI Count	W
40481	Firmware version (low word)	R
40482	Firmware version (high word)	R

40483	Module name (low word)	R
40484	Module name (high word)	R
30001	DI Count Value	R
00270	Host watch dog timeout status, write 1 to clear host watchdog timeout status.	R/W
40488	Modbus reponse delay time in ms, valid range:0~30	R/W
40492	Host watchdog timeout count,write 0 to clear	R/W
00260	Modbus host watchdog mode 0 : same as I-7000 1 : can use AO and DO command to clear host watchdog timeout status	R/W
00261	1 : enable, 0 : disable host watchdog	R/W
40489	Host watchdog timeout value, 0~255, in 0.1s	R/W
00257	Protocol, 0: DCON, 1: Modbus	R/W
00258	Modbus Protocol Option, 0: RTU, 1:ASCII	R/W
10273	Reset status, 1: first read after powered on, 0 : not the first read after powered on	R
312345, 412345	Informs all modules that the host is OK	R

## 1. Modbus RTU Function Description :

(0xxxx) : 0x01、0x05、0x0F Function code

(1xxxx) : 0x02 Function code

(3xxxx) : 0x04 Function code

(4xxxx) : 0x03、0x03、0x10 Function code

## 2. Address mapping of PLC(Base 1).

Function code	Description	Section
0x01	Read coils	3.1
0x02	Read discrete inputs	3.2
0x03	Read multiple registers	3.3
0x04	Read multiple input registers	3.4
0x05	Write single coils	3.5
0x06	Write multiple registers	3.6
0x0F	Write multiple coils	3.7
0x46	Read/Write module settings	3.8

If the function specified in the message is not supported, then the module responds as follows. Address mapping of Protocol(Base 0).

### Error Response

00	Address	1 Byte	1 to 247
01	Function code	1 Byte	Function code + 0x80
02	Exception code	1 Byte	01

If a CRC mismatch occurs, the module will not respond.

### 3.1 01(0x01) Read coils

This function code is used to read the current digital output readback value.

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x01
02-03	Starting channel numbers or address mapping	2 Byte	<ul style="list-style-type: none"><li>● DO : 0x0000-0x001F</li><li>● DI : 0x0020-0x003F</li><li>● DI/O Latch High : 0x0040-0x005F</li><li>● DI/O Latch Low : 0x0060-0x007F</li><li>● Safe Value : 0x0080-0x009F</li><li>● Power On Value : 0x00A0-0x00BF</li><li>● Reads WDT Mode : 0x0103</li><li>● Reads WDT Enable : 0x0104</li><li>● Reads WDT Status : 0x010D</li><li>● Reads Protocol : 0x0100</li><li>● Reads Reset Status : 0x0110</li></ul>
04-05	Output channel number or bit count	2 Byte	0x0001-0x0020 (Bit count)

## **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x01
02	Byte Count	1 Byte	Byte count of response (B=(bit count + 7)/8)
03	Bit values	B Byte	(Bit values)

## **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x81
02	Exception code	1 Byte	Refer to Modbus standard for more details

## **Examples:**

Command:05 01 00 00 00 05 [ FD 8D ]

Response:05 01 01 1F 11 70

Reads digital output value

[Command]

Byte 2~3 : 00 00 (starting channel numbers)

Byte 4~5 : 00 05 (output channel number)

Byte 6~7 : FD 8D(CRC)

[Response]

Byte 1 : 01 (Function Code)

Byte 2 : 01 (Byte count of response)

Byte 3 : 1F (DO0~DO3 Value)

Byte 4~5 : 11 70 (CRC)

Command:05 01 00 40 00 04 [ 3D 99 ]

Response:05 01 01 07 11 7A

Reads digital input high latch

Command:05 01 01 03 00 01 [ 0D B2 ]

Response:05 01 01 00 50 B8

Reads modbus host watchdog's mode

Command:05 01 01 10 00 01 [ FC 77 ]

Response:05 01 01 01 91 78

Reads reset status of modules

Command:05 01 01 04 00 01 [ BC 73 ]

Response:05 01 01 00 50 B8

Reads host watchdog setting (enabled or not)

Command:05 01 01 0D 00 01 [ 6C 71 ]

Response:05 01 01 00 50 B8

Resets the host watchdog timeout status of a module

Command:05 01 01 00 00 09 [ 6C 71 ]

Response:05 01 02 01 00 49 AC

Reads protocol of module

## **Supported modules :**

**M-7065**

Valid starting channel	DO : 0x0000~0x0004
	DI : 0x0020~0x0023
	High Latched DI Channel : 0x0040~0x0043
	Low Latched DI Channel : 0x0060~0x0063
	Safe Value : 0x0080~0x0084
	Power On Value : 0x00A0~0x00A4

## 3.2 02(0x02) Read discrete inputs

This function code is used to read the current digital input value.

### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x02
02-03	Starting channel numbers	2 Byte	● DI : 0x0000-0x001F
04-05	Input channel number	2 Byte	0x0001-0x0020 (Bit count)

### Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x02
02	Byte Count	1 Byte	Byte count of response (B=(bit count + 7)/8)
03	Bit values	B Byte	Bit values

### Error Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x82
02	Exception code	1 Byte	Refer to Modbus standard for more details

## **Examples:**

Command:05 02 00 00 00 04 [ 78 4D ]

Response:05 02 01 0F E0 BC

Reads digital iinput value

[Command]

Byte 2~3 : 00 00 (starting channel numbers)

Byte 4~5 : 00 04 (input channel number)

Byte 6~7 : 78 4D (CRC)

[Response]

Byte 1 : 02 (Function Code)

Byte 2 : 01 (Byte count of response)

Byte 3 : 0F (DI0~DI3 Value)

## **Supported modules :**

M-7065

Valid Starting channel	DI : 0x0000~0x0003
---------------------------	--------------------

### 3.3 03(0x03) Read multiple registers

This function code is used to read the current digital input counter value.

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x03
02-03	Starting channel numbers or address mapping	2 Byte	<ul style="list-style-type: none"><li>● Reads DI count value : 0x0000-0x001F</li><li>● Reads module address:0x01E4</li><li>● Reads firmware version:0x01E0</li><li>● Reads module name:0x01E2</li><li>● Reads modbus delay response time : 0x01E7</li><li>● Reads timeout count : 0x01EB</li><li>● Reads timeout value : 0x01E8</li><li>● Host OK : 0x3038</li></ul>
04-05	Input channel number	2 Byte	0x0001-0x0020 (Word count)

## **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x03
02	Byte Count	1 Byte	Byte count of response (B=2 * word count)
03~	Register values	(B*2) Byte	Register values

## **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x83
02	Exception code	1 Byte	Refer to Modbus standard for more details

## **Examples:**

Command:05 03 01 E2 00 02 [ 64 45 ]

Response:05 03 04 70 65 00 00 B5 2C

Reads module's name

[Command]

Byte 2~3 : 01 E2 (Address mapping)

Byte 4~5 : 00 02 (Word count)

[Response]

Byte 2 : 04 (Byte count of response)

Byte 3~6 : 70 65 00 00, 70 65(Low word of module name), 00 00(High word of module name)

Command:05 03 01 E4 00 01 [ C4 45 ]

Response:05 03 02 00 05 89 87

Reads address of modules

Command:01 03 00 00 00 08 [ 44 0C ]

Response:01 03 10 00 15 00 15 00 15 00 15 00

15 00 15 00 15 2D 56

Reads DI count of modules

Command:05 03 01 E7 00 01 [ 34 45 ]

Response:05 03 02 00 00 49 84

Reads modbus response delay time

Command:05 03 01 EB 00 01 [ F4 46 ]

Response:05 03 02 00 00 49 84

Reads the host watchdog timeout value of a module

Command:00 03 30 38 00 01 [ 0B 16 ]

Response:No response

Informs all modules that the host is OK

## **Supported modules :**

**M-7065**

Valid starting channel	DI Count Value : 0x0000-0x0003
---------------------------	--------------------------------

### 3.4 04(0x04) Read multiple registers

This function code is used to read the current digital input counter value.

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x04
02-03	Starting channel numbers or address mapping	2 Byte	<ul style="list-style-type: none"><li>● Reads DI count value : 0x0000-0x001F</li><li>● Reads module address:0x01E4</li><li>● Reads firmware version:0x01E0</li><li>● Reads module name:0x01E2</li><li>● Reads modbus delay response time : 0x01E7</li><li>● Reads timeout count : 0x01EB</li><li>● Reads timeout value : 0x01E8</li><li>● Host OK : 0x3038</li></ul>
04-05	Input channel number	2 Byte	0x0001-0x0020 (Word count)

## **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x04
02	Byte Count	1 Byte	Byte count of response (B=2 * word count)
03~	Register values	(B*2) Byte	Register values

## **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x83
02	Exception code	1 Byte	Refer to Modbus standard for more details

## **Examples:**

Command:05 04 01 E2 00 02 [ D1 85 ]

Response:05 04 04 20 60 00 00 B5 9A

Reads module's name

Command:01 04 01 E4 00 01 [ 70 01 ]

Response:01 04 02 00 01 78 F0

Reads address of modules

Command:01 03 00 00 00 02 [ C4 0B ]

Response:01 03 04 00 15 00 15 2A 38

Reads DI count of modules

Command:05 04 01 E7 00 01 [ 81 85 ]

Response:05 04 02 00 00 48 F0

Reads modbus response delay time

Command:05 04 01 EB 00 01 [ 41 86 ]

Response:05 04 02 00 00 48 F0

Reads the host watchdog timeout value of a module

Command:00 04 30 38 00 01 [ BE D6 ]

Response:No response

Informs all modules that the host is OK

### **Supported modules :**

M-7065

Valid starting channel	DI Count Value : 0x0000-0x0005
---------------------------	--------------------------------

### 3.5 05(0x05) Write single coils

This function code is used to write the digital output value.

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x05
02-03	Output channel numbers	2 Byte	<ul style="list-style-type: none"><li>● DO : 0x0000-0x001F</li><li>● Clear DI count value : 0x0200-0x021F</li><li>● Safe Value : 0x0080-0x009F</li><li>● Power On Value : 0x00A0-0x00BF</li><li>● Clear Latched : 0x0107</li><li>● Set WDT Mode : 0x0103</li><li>● Set WDT Enable : 0x0104</li><li>● Clear WDT Status : 0x010D</li></ul>
04-05	Output value	2 Byte	A value 0xFF00 sets the output to ON A value 0x0000 sets the output to OFF

## **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x05
02-03	Output channel numbers	2 Byte	The value is the same as byte 02 and 03 of the Request
04-05	Output value	2 Byte	The value is the same as byte 04 and 05 of the Request

## **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x85
02	Exception code	1 Byte	Refer to Modbus standard for more details

## **Examples:**

Command:05 05 00 83 FF 00 [ 7C 56 ]

Response:05 05 00 83 FF 00 7C 56

Sets DO3 power-on value to on

Command:05 05 02 00 FF 00 [ 8C 06 ]

Response:05 05 02 00 FF 00 8C 06

Clear digital input count

Command:05 05 00 02 FF 00 [ 2C 7E ]

Response:05 05 00 02 FF 00 2C 7E

Sets DO2 to on

Command:05 05 01 07 FF 00 [ 3D 83 ]

Response:05 05 01 07 FF 00 3D 83

Clear DIO latch of modules

Command:05 05 01 03 FF 00 [ 7C 42 ]

Response:05 05 01 03 FF 00 7C 42

Sets host watchdog mode (Mode 1)

Command:05 05 01 03 00 00 [ 3D B2 ]

Response:05 05 01 03 00 00 3D B2

Sets host watchdog mode (Mode 0)

Command:05 05 01 04 FF 00 [ CD 83 ]

Response:05 05 01 04 FF 00 CD 83

Sets host watchdog enable

Command:05 05 01 0D FF 00 [ 1D 81 ]

Response:05 05 01 0D FF 00 1D 81

Clear host watchdog status

Command:05 05 01 00 FF 00 [ 8C 42 ]

Response:05 05 01 00 FF 00 8C 42

Sets protocol (Bit 0)

[Command]

Byte 2~3 : 01 00 [Protocol (Bit 0)]

Byte 4~5 : FF 00

Protocol (bit1~bit0)

11	01	00
Modbus ASCII	Modbus RTU	DCON

Command:05 05 01 01 FF 00 [ DD 82 ]

Response:05 05 01 01 FF 00 DD 82

Sets protocol (Bit 1)

[Command]

Byte 2~3 : 01 01 [Protocol (Bit 1)]

Byte 4~5 : FF 00

Protocol (bit1~bit0)

11	01	00
Modbus ASCII	Modbus RTU	DCON

## Supported modules :

### M-7065

Valid starting channel	DO : 0x0000~0x0005
	Clear DI Count Value : 0x0200-0x0203
	Safe Value : 0x0080~0x0085
	Power On Value : 0x00A0~0x00A5

### 3.6 06(0x06) Write multiple registers

This function code is used to set the settings of the module.

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x06
02-03	Address mapping	2 Byte	<ul style="list-style-type: none"><li>● Sets modbus reponse delay time : 0x01E7</li><li>● Clear watchdog timeout count : 0x01EB</li><li>● Sets watchdog timeout value : 0x01E8</li></ul>
04-05	Register value	2 Byte	Register value

#### Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	06
02-03	Address mapping	2 Byte	The value is the same as byte 02 and 03 of the Request
04-05	Register value	2 Byte	Register value

#### Error Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x86
02~	Exception code	n Byte	Refer to Modbus standard for more details

## **Examples:**

Command:05 06 01 E7 00 10 [ 38 49 ]

Response:05 06 01 E7 00 10 38 49

Sets modbus response delay time (0~30,in 1ms)

[Command]

Byte 2~3 : 01 E7 (Address mapping)

Byte 4~5 : 00 10 (delay 16 ms)

[Response]

Byte 1 : 06 (Function Code)

Command:05 06 01 E8 00 C8 [ 08 10 ]

Response:05 06 01 E8 00 C8 08 10

Sets host watchdog timeout value (0~255,in 0.1s)

Command:09 06 01 EB 00 00 [ F9 4A ]

Response:09 06 01 EB 00 00 F9 4A

Clear host watchdog timeout count

### **3.7 15(0x0F) Write multiple coils**

This function code is used to write the digital output value.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x0F
02-03	Starting channel numbers	2 Byte	<ul style="list-style-type: none"><li>● DO : 0x0000-0x001F</li><li>● Clear DI count value : 0x0200-0x021F</li><li>● Safe Value : 0x0080-0x009F</li><li>● Power On Value : 0x00A0-0x00BF</li></ul>
04-05	Output channel number	2 Byte	0x0001-0x0020 (Bit count)
06	Byte Count	1 Byte	B=(bit count + 7)/8
07	Output value	2 Byte	A bit corresponds to a channel. When the bit is 1 it denotes that the value of the channel that was set is ON. If the bit is 0 it denotes that the value of the channel that was set is OFF.

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x0F
02-03	Starting channel numbers	2 Byte	The value is the same as byte 02 and 03 of the Request
04-05	Input channel number	2 Byte	0x0001-0x0020

## Error Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x8F
02	Exception code	1 Byte	Refer to Modbus standard for more details

## Examples:

Command:05 0F 00 A1 00 03 01 07 [ 72 BF ]

Response:05 0F 00 A1 00 03 45 AC

Sets Safe value (DO1-DO3)

Command:05 0F 00 00 00 03 01 FF [ CE E4 ]

Response:05 0F 00 00 00 03 14 4E

Sets DO value (DO0-DO3)

Command:01 0F 02 00 00 08 01 FF [ BF 37 ]

Response:01 0F 02 00 00 08 55 B5

Clear DI count (DI0-DI7)

## **Supported modules :**

**M-7065**

Valid starting channel	DO : 0x0000~0x0005
	Clear DI Count Value : 0x0200-0x0203
	Safe Value : 0x0080~0x0085
	Power On Value : 0x00A0~0x00A5

## 3.8 46(0x46) Read/Write module settings

This function code is used to read the settings of the module or change the settings of the module. The following sub-function codes are supported

Sub-Function code	Description	Section
00 (0x00)	Read the module name	3.8.1
04 (0x04)	Set the module address	3.8.2
05 (0x05)	Read the communication settings	3.8.3
06 (0x06)	Set the communication settings	3.8.4
32 (0x20)	Read the firmware version	3.8.5
33 (0x21)	Set the DI count edge	3.8.6
34 (0x22)	Read DI count edge setting value	3.8.7
39 (0x27)	Set the DO power-on value	3.8.8
40 (0x28)	Read the DO power-on value	3.8.9
41 (0x29)	Set the DI/O active status	3.8.10
42 (0x2A)	Read the DI/O active status	3.8.11
53 (0x35)	Read the response delay time	3.8.12
54 (0x36)	Set the response delay time	3.8.13

## Error Response

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

### **3.8.1 00(0x00) Read Module Name**

This sub-function code is used to read the name of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x00

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x00
03-06	Module name	4 Byte	0x00 0x70 0x65 0x00 for M-7065

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

#### **Examples:**

Command:05 46 00 [ 53 A1 ]

Response:05 46 00 00 70 65 00 6A 2D

### **3.8.2 04(0x04) Set Module Address**

This sub-function code is used to set the address of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x04
03	Address	1 Byte	1-247
04-06	Reserved	3 Byte	0x00 0x00 0x00

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x00
03	Address	1 Byte	0 : OK Others : Error
04-06	Reserved	3 Byte	0x00 0x00 0x00

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

### 3.8.3 05(0x05) Read Communication

This sub-function code is used to read the communication protocol settings of a module

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x05
03	Reserved	1 Byte	0x00

#### Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x05
03	Protocol support	1 Byte	0 ,1 : Modbus RTU 3 : Modbus RTU/ASCII
04	Baud Rate	1 Byte	0x0A
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0 : N81 1 : N82 2 : E81 3 : O81
07	Reserved	1 Byte	0x00
08	Protocol Mode	1 Byte	0 : DCON 1 : Modbus RTU 3 : Modbus ASCII
09-10	Reserved	2 Byte	0x00 0x00

## Error Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

## Examples:

Command:02 46 05 00

Response:02 46 05 03 0A 00 00 00 01 00 00 6B 12

Reads protocol

### 3.8.4 06(0x06) Set Communication

This sub-function code is used to set the communication protocol of a module.

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x06
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0x0A
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0 : N81 1 : N82 2 : E81 3 : O81
07	Reserved	1 Byte	0x00
08	Protocol Mode	1 Byte	0 : DCON 1 : Modbus RTU 3 : Modbus ASCII
09-10	Reserved	2 Byte	0x00 0x00

#### Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x06
03	Reserved	1 Byte	0x00
04	Baud Rate	1 Byte	0 : OK

			Others : Error
05	Reserved	1 Byte	0x00
06	Parity	1 Byte	0 : OK Others : Error
07	Reserved	1 Byte	0x00
08	Protocol Mode	1 Byte	0 : OK Others : Error
09-10	Reserved	2 Byte	0x00 0x00

### Error Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

### Examples:

Command:02 46 06 00 09 00 02 00 03 00 00 [ D4  
F7 ]

Response:02 46 06 00 00 00 00 00 00 00 C4 37  
Sets the configuration of module 02.

### **3.8.5 32(0x20) Read Firmware Version**

This sub-function code is used to read the firmware version information of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x20

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x20
03	Major version	1 Byte	0x00-0xFF
04	Minor version	1 Byte	0x00-0xFF
05	Build version	1 Byte	0x00-0xFF

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

#### **Examples:**

Command:01 46 20 [ 52 79 ]

Response:05 46 20 02 00 00 23 81

### 3.8.6 33(0x21) Set Digital Input Count Edge

This sub-function code is used to set the digital input counter trigger edge value of a module.

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x21
03	Edge setting value	1 Byte	0x00-0xFF (DI0-DI7)
04	Edge setting value	1 Byte	0x00-0xFF (DI8-DI15)
05	Edge setting value	1 Byte	0x00-0xFF (DI16-DI23)
06	Edge setting value	1 Byte	0x00-0xFF (DI24-DI31)

\*1 = rising edge, 0 = falling edge. For example 0x03 denotes that channels 0~1 are set as rising edge and channels 2~3 are set as falling edge

#### Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x21
03	Edge setting value	1 Byte	0 : OK Others : Error

#### Error Response:

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

**Examples:**

Command:01 46 21 FF [ B8 1D ]

Response:01 46 21 00 F8 5D

Sets digital input count edge (Rising edge)

**Examples:**

Command:01 46 22 [ 92 79 ]

Response:01 46 22 FF B8 ED

Reads digital input count edge

### **3.8.7 34(0x22) Read Digital Input Count Edge**

This sub-function code is used to read the digital input counter trigger edge value of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x22

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x22
03	Edge setting value	1 Byte	0x00-0xFF (DI0-DI7)
04	Edge setting value	1 Byte	0x00-0xFF (DI8-DI15)
05	Edge setting value	1 Byte	0x00-0xFF (DI16-DI23)
06	Edge setting value	1 Byte	0x00-0xFF (DI24-DI31)

\*1 = rising edge, 0 = falling edge. For example 0x03 denotes that channels 0~1 are set as rising edge and channels 2~3 are set as falling edge

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

**Examples:**

Command:05 46 21 3F [ B9 7D ]

Response:05 46 21 00 F9 6D

Sets DI0-DI5 count edge (Rising edge)

**Examples:**

Command:05 46 22 [ D3 B8 ]

Response:05 46 22 3F B9 8D

Reads digital input count edge

### **3.8.8 39(0x27) Set Power-on Value**

This sub-function code is used to set the power-on value of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x27
03	Power-on Value	1 Byte	0x00-0xFF (DOI0-DO7)
04	Power-on Value	1 Byte	0x00-0xFF (DO8-DO15)
05	Power-on Value	1 Byte	0x00-0xFF (DO16-DO23)
06	Power-on Value	1 Byte	0x00-0xFF (DO24-DO31)

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x27
03	Power-on Value	1 Byte	0 : OK Others : Error

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

**Examples:**

Command:05 46 27 0F [ BA C9 ]

Response:05 46 27 00 FA CD

Sets D00-D03 power-on value

**Examples:**

Command:05 46 28 [ 53 BF ]

Response:05 46 28 0F BF 39

Reads power-on value

### **3.8.9 40(0x28) Read Power-on Value**

This sub-function code is used to read the power-on value of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x28

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x28
03	Power-on Value	1 Byte	0x00-0xFF (DOI0-DO7)
04	Power-on Value	1 Byte	0x00-0xFF (DO8-DO15)
05	Power-on Value	1 Byte	0x00-0xFF (DO16-DO23)
06	Power-on Value	1 Byte	0x00-0xFF (DO24-DO31)

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

### 3.8.10 40(0x29) Set DI/O Active Status

This sub-function code is used to set the DI/O active states of a module.

#### Request

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x29
03	DI/O Active Status	1 Byte	0x00-0x03

7	6	5	4	3	2	1	0
Reserved						OAS	IAS

Key	Description
OAS	DO Active Status 0: output value 1 for relay active output value 0 for relay inactive 1: output value 0 for relay active output value 1 for relay inactive
IAS	DI Active Status 0: input value 1 for non-signal or the low voltage; input value 0 for high voltage 1: input value 0 for non-signal or the low voltage; input value 1 for high voltage

(DI/O Active Status)

## **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x29
03	DI/O Active Status	1 Byte	0 : OK Others : Error

## **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

## **Examples:**

Command:01 46 29 02 [ 7E 5C ]

Response:01 46 29 00 FF 9D

Sets the DI/O active states

## **Examples:**

Command:01 46 2A [ 93 BF ]

Response:01 46 2A 02 7E AC

Reads the DI/O active states

### **3.8.11 42(0x2A) Read DI/O Active Status**

This sub-function code is used to read the DI/O active states of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x2A

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x2A
03	DI/O Active Status	1 Byte	0x00-0x03

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

#### **Examples:**

Command:01 46 29 02 [ 7E 5C ]

Response:01 46 29 00 FF 9D

Sets the DI/O active states

Command:01 46 2A [ 93 BF ]

Response:01 46 2A 02 7E AC

Reads the DI/O active states

7	6	5	4	3	2	1	0
Reserved						OAS	IAS

Key	Description
OAS	DO Active Status 0: output value 1 for relay active output value 0 for relay inactive 1: output value 0 for relay active output value 1 for relay inactive
IAS	DI Active Status 0: input value 1 for non-signal or the low voltage; input value 0 for high voltage 1: input value 0 for non-signal or the low voltage; input value 1 for high voltage

(DI/O Active Status)

### **3.8.12 53(0x35) Read the Response Delay Time**

This sub-function code is used to read modbus response delay time of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x35

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x35
03	Response Delay Time	1 Byte	(Data)

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

#### **Examples:**

Command:01 46 36 1E [ 77 A5 ]

Response:01 46 36 1E 77 A5

Sets modbus response delay time (30ms)

**Examples:**

Command:01 46 35 [ D2 77 ]

Response:01 46 35 1E 77 55

Reads modbus response delay time (30ms)

### **3.8.13 54(0x36) Set the Response Delay Time**

This sub-function code is used to set modbus response delay time of a module.

#### **Request**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x36
03	Response Delay Time	1 Byte	(Data)

#### **Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0x46
02	Sub-Function code	1 Byte	0x36
03	Response Delay Time	1 Byte	(Data)

#### **Error Response:**

00	Address	1 Byte	1-247
01	Function code	1 Byte	0xC6
02	Exception code	1 Byte	Refer to Modbus standard for more details

## 4. Modbus ASCII Protocol

Using Modbus ASCII Protocol, all command are coded in hexadecimal values, represented with readable ASCII characters. Only the characters 0...9 and A...F are used for coding.

Using Modbus ASCII Protocol, characters are used to start and end a frame. The Leading Character ':' is used to flag the start of a command and each command is ended with a CR•LF combination. The LRC characters are appended to the command preceding the CR•LF characters.

### Command Format:

Leading Character	Module Address	Command	[LRC]	CR	LF
-------------------	----------------	---------	-------	----	----

### Response Format:

Leading Character	Module Address	Data	[LRC]	CR	LF
-------------------	----------------	------	-------	----	----

### LRC Calculation:

All characters except for Leading Character (:) and delimiter (CR•LF) are added with a carry being discarded.

Total value is converted to binary notation, is converted to 2's complements, then to hexadecimal figures, that is, LRC.

**Example :**

Modbus RTU => 01 46 00 [ 12 60 ]

Modbus ASCII => :014600B9(CR • LF)

(1.) hexadecimal = 01h+46h+00h = 47h

(2.) 2's complement: = B9h (LRC)

# A. Appendix

## A.1 INIT Mode

The M-7065 module has a build-in EEPROM to store configuration information such as module address, type code, Baud Rate, etc. Occasionally, the configuration of a module may be forgotten and there are no visual indications of the configuration of the module. It is difficult to communicate with the module when the configuration of the module is unknown. To help avoid this problem, the M-7065 has a special mode called “INIT mode”. When the module is powered on in “INIT mode” the configuration of the module is reset as follows, allowing it to be operated as normal.

1. Address: 00
2. Baud Rate: 9600pbs,n,8,1
3. No checksum
4. Protocol: DCON .

The configuration information stored in the EEPROM is not changed and can be read by sending the \$002(CR) command at 9600bps.

There are commands that require the module to be in INIT mode. They are:

1. %AANNTTCCFF when changing the Baud Rate and checksum settings. See Section 2.1 for details.
2. \$AAPN, see Section 2.18 for details.

## A.2 Dual Watchdog operation

### Dual Watchdog = Module Watchdog + Host Watchdog

The Module Watchdog is a hardware reset circuit that monitors the operating status of the module. While working in harsh or noisy environments, the module may be shut down by external signals. The circuit allows the module to work continuously without disruption.

The Host Watchdog is a software function that monitors the operating status of the host. Its purpose is to prevent problems due to network/communication errors or host malfunctions. When a host watchdog timeout occurs, the module will reset all outputs to a safe state in order to prevent any erroneous operations of the controlled target

The M-7065 DIO module include an internal Dual Watchdog, making the control system more reliable and stable.

## A.3 Reset Status

The reset status of a module is set when the module is powered-on or when the module is reset by the module watchdog. It is cleared after the responding of the first \$AA5 command. This can be used to check whether the module had been reset. When the \$AA5 command responds that the reset status is cleared, that means the module has not been reset since the last \$AA5 command was sent. When the \$AA5 command responds that the reset status is set and it is not the first time \$AA5 command is sent, it means the module has been reset and the digital output value had been changed to the power-on value.

## A.4 Digital Output

Besides setting by the set digital output commands, the digital outputs can be set by two other conditions.

When the host watchdog is enabled and a host watchdog timeout occurs, the “safe value” is loaded into the digital output ports. The set digital output commands have no effect on the digital output ports until the host watchdog timeout status is cleared. The host watchdog timeout status is saved in the EEPROM. The status is not changed even after power-on reset. It can be cleared only by the reset host watchdog timeout status command ~AA1. See Section A.2 for host watchdog details.

When the module is powered on and the host watchdog timeout status is cleared, the “power-on value” is loaded into the digital output ports. If the host watchdog timeout status is not cleared on power-on, then the safe value is loaded into the digital output ports.

Both the safe value and power-on value are set by the ~AA5V command. Refer to Section 2.32 for details.

## A.5 Latched Digital Input

The M-7065 DIO module provide commands to read the latched high digital input and latched low digital input status. Following is an example to show the usefulness of the latched digital input. When we want to read the key stroke of a key switch which is connected to the digital input channel of a module, the input signal of the key stroke is a pulse signal as shown in the following figure.



If we just use the read digital input status command to read the signal and we cannot send the command during the B period due to some reasons, then we will lose the key stroke information. However, with the read latched digital input command, we can still get the key stroke information even we are not able to send command in B period. For details of the read latched digital input command, please refer to Sections 2.15.